

# Multi-sensor data fusion and nonlinear programming-based path prediction for escaping from engagement in combat

Enver Nurullah GÖKAL  and Ufuk SAKARYA 

One of the most important factors that bring success in modern warfare is to show air superiority. Unmanned aerial vehicles (UAVs) have now become an essential component of military air operations. UAVs can be operated in two ways: by pilots from remote control stations or by flying autonomously. Under the condition of disconnection from the control station, UAVs have trouble maintaining navigation and maneuverability. By applying multi-sensor data fusion, an escape path prediction algorithm was developed and presented as an engagement escape method in this study. To develop the algorithm for prediction of the optimal escape route, data from various sensors are collected and processed under the influence of noise. The data from the distance and angle sensors are interpreted in the Extended Kalman Filter and estimations are made. The instant optimal escape route is created by applying the constrained optimization method on the estimations made. The main motivation of this study is developing a deterministic-based method to get the certification of it in aviation. Therefore, instead of stochastic-based learning approaches, a deterministic approach is preferred. Nonlinear programming is used as the constraint optimization method because the constraints and objective function are nonlinear. In the selected scenarios, it can be seen in the simulation results that the proposed method shows a promising result in terms of escape from engagement.

**Key words:** escaping from engagement, nonlinear programming, multi-sensor data fusion

## 1. Introduction

Among the factors that bring success in modern warfare, one of the most important factors is to show air superiority. When an aircraft is in a man-made

---

Copyright © 2024. The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License (CC BY-NC-ND 4.0 <https://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits use, distribution, and reproduction in any medium, provided that the article is properly cited, the use is non-commercial, and no modifications or adaptations are made

E.N. Gököl (corresponding author, e-mail: [envernurullah.gokal@tai.com.tr](mailto:envernurullah.gokal@tai.com.tr)) is with Turkish Aerospace Industries, İstanbul, Republic of Türkiye.

U. Sakarya (e-mail: [usakarya@yildiz.edu.tr](mailto:usakarya@yildiz.edu.tr)) is with YILDIZ Technical University, Faculty of Applied Sciences, Department of Aviation Electrics and Electronics, and with Faculty of Electrical and Electronics, Department of Electronics and Communication Engineering, İstanbul, Republic of Türkiye.

Received 14.02.2023. Revised 11.03.2024.

hostile environment, having survivability features ready to use and hidden in the aircraft is of great importance in terms of providing air dominance. To make combat aircraft have better maneuverability in engagement conditions, it is focused on a decision-making algorithm in this study. In military jargon, the term “engagement” is used to refer to the interaction between the two parties. The attacking side initiates the engagement to perform a mission. When the mission ends with failure or success, the engagement ends [9]. It is important for an aircraft to perform fast decision making based on the maneuverability of the aircraft and the opponent’s position to successfully escape the engagement. The main scenario in this study is a dogfight engagement with an attacking aircraft. The aim is to ensure that the attacked makes its escape from engagement in the most optimized way. In the discussed scenarios in this study, both the attacked and the attacking parties are unmanned aerial vehicles (UAVs). The attacking side also may be Surface-to-Air Missile (SAM) or Air-to-Air Missile (AAM), instead of a UAV.

The most basic feature of UAVs is that there is no human inside the UAV; thus, avoiding the risk of pilots being harmed in battle [8]. UAVs have now become an indispensable element of air operations in the military. Due to the absence of a human pilot operating the aircraft, nations can perform military operations with less risk using UAVs. In addition, UAVs, with their functions, provide great efficiency contribution to armies in military operations. Some of the uses of UAVs, which can be used in a wide variety of ways in war, are as follows; surveillance, assigning a target to other armed systems, attacking the target by itself. UAVs can attack fixed or moving targets [12]. Medium Altitude Long Endurance (MALE) Anka UAV produced by Turkish Aerospace Industries can serve as an example of a UAV [1]. Anka is the military UAV designed and manufactured to perform surveillance, reconnaissance, target detection and target recognition tasks.

Movement control of UAVs can be performed in two ways; UAVs are controlled by pilots from remote control stations or can fly autonomously. In addition to the remote-control station, communication satellites and GNSS satellites can also be used in flight control. When a UAV controlled from a remote-control station is disconnected from the control station, it should be able to continue its flight autonomously in order to avoid a catastrophic result [6]. The basic methods that provide autonomous flight in UAVs are as follows; artificial neural networks, machine learning and real-time path planning with artificial intelligence or numerical methods [7]. Other works based on different approaches are present for this subject. A guidance and navigation system for Remotely Piloted Aircraft Systems (RPAS) based on particle filter has proposed by Capello et al. [5]. In the proposed method, particle filter-based estimation method was examined, and a comparison is made between particle filter, Extended Kalman Filter and

Unscented Kalman Filter. An autonomous decision-making algorithm with 14 different maneuvering options for unmanned combat aircraft (UCAV) has been proposed by López and Zbikowski [25]. A score equation considering external constraints is used to evaluate each decision. Zhang et al. [23] proposed a method of sequential convex programming for path planning to be used in UAVs. When the problem is a nonlinear control problem which is non-convex, a method for optimization by conversion of the non-convex problem to a convex problem or series of convex problems is proposed.

Tisdale et al. [30] proposed a camera-based estimation and path prediction method for a search and locating task for a group of UAVs. In the proposed method, recursive Bayesian estimation was used as the estimation method, and non-convex constrained optimization was used in the path planning stage. Bortoff [2] proposed a UAV path planning algorithm. Aim of the proposed two-stage path planning algorithm is to determine an optimal path for UAVs by making a trade-off between stealth and path length in areas with radar. Jiang and Liang [15] have proposed a case-based path planning algorithm with a standoff distance for autonomous UAVs. In the proposed method, target trajectory estimation has been made with quadratic functions. For target localization, a nonlinear least squares estimation method is used. Finally, a case-based decision-making algorithm stepped in to accomplish the path planning with a standoff distance. In the proposed method, sensor noise is considered when using the measured values of the sensors. A new case-based guidance method has been proposed to make a balanced trade-off between the optimal path planning method and real-time performance.

Yang et al. [33] proposed a path planning method using passive detection system for target detection. The use of radar in the target detection system makes it easier for the aircraft to reveal its position by other systems, so a method using a passive detection system as a target detection system has been proposed. In the proposed method, Partially Observable Markov Decision Process is used as the decision-making algorithm. Kang et al. [16] proposed a Kalman filter algorithm for UAV path planning. In this study, it is aimed to eliminate the negativities caused by the uncertainties in the flight environment. With a Kalman filter-based module developed by taking into account sensor noises, GPS sensor noise and model noise, threats were modeled and a safe distance for the UAV has attempted to be determined. Wu et al. [32] proposed a path planning method which uses a Kalman filter-based prediction algorithm for collision avoidance in UAV groups. To eliminate the inconsistencies coming from the noise caused by the communication of many UAVs in a dynamic environment, a Kalman filter based estimation method has been proposed.

Luo et al. [19] proposed a position estimation and collision avoidance method for UAVs. Due to the colored noise received in the signal strength measurement

coming from the communication module, the distance estimation was performed using the colored noise model in the Extended Kalman Filter. Mao et al. [20] proposed an Extended Kalman Filter design for position estimation for UAVs. The proposed design concerns situation where the UAVs temporarily lost their GPS connection. In the scenario created, a group of UAVs cooperating with each other is considered.

In this study, it is aimed developing a path planning algorithm so that the UAV can survive engagement on its own when the connection between the UAV and the ground control station is broken during engagement. It is significant for a UAV to use its autonomous flight features to prevent any catastrophic results in an engagement when the connection link from ground control station or satellite has cut down. These autonomous flight features include optimal escape path prediction. When the scenario stated above is occurred, UAV shall generate its path using optimal path prediction algorithm to escape from engagement. For this purpose, in the optimal escape path prediction algorithm to be developed in this study, data from different sensors are processed and combined under the influence of noise. The main motivation of this study is developing a deterministic-based method to get the certification of it in aviation. Therefore, instead of stochastic-based learning approaches, a deterministic approach is preferred.

In addition, in this study, the standard deviation values of the measurement noise representing the noise in the sensors and the process noise representing the noise in the environment are applied as user inputs to the Extended Kalman Filter model used for estimation. It is aimed to observe how the Extended Kalman Filter affects the operation of this method at different noise levels by making simulations at different noise values. This observation is important to determine the impact of the proposed method on the resource utilization of the UAV. As the constraint optimization method of this study, the nonlinear programming method [3] is used. In this study, it is also aimed to observe the effects of the following parts of the nonlinear programming algorithm, objective function, linear and nonlinear equality, and inequality constraints on the optimal escape path of the UAV. While constraint optimization is made, weight values of the distance and the angle have a decisive effect on the planned escape path. In other words, the weight values of the distance and the angle effects the resource utilization of the UAV.

The method presented in this article is an Extended Kalman Filter and nonlinear programming-based UAV navigation and guidance method. As a first step in this method, data from different sensors are combined in the Extended Kalman Filter to estimate the position of the enemy aircraft. The two sensors used are the range and angle sensors. In the second step of the method, after estimating the position and direction of the enemy aircraft, the constraint optimization is performed using the nonlinear programming method. The constraints are made

on the escaping UAV's coordinate axes and the maneuverability of the UAV changes in the pitch, yaw and roll axes. Nonlinear programming is applied on the escaping UAV's coordinate axes and an appropriate optimization is made for the axes. The optimal solution found according to the UAV coordinate axes is applied to the original coordinate axis by three-dimensional transformation. In summary, an Extended Kalman Filter and nonlinear programming-based escape path prediction algorithm has been developed.

The preliminary version of this study is available in [13]. The main difference between [13] and this article is inclusion of the angle constraints in the nonlinear programming. The angular constraints are important for the model to realize more realistic scenarios. Moreover, according to this improvement in the model, the new scenarios are simulated and demonstrated in this article.

The paper is organized as follows. In Section 2, necessary background information about the methods and algorithms used in this study is given. In Section 3, the proposed method is presented. In Section 4, the three scenarios run on the simulation are described first. Then, results of the simulation are examined for the three different scenarios separately. In addition, a ground truth calculation is described as well. Conclusion is presented in Section 5.

## 2. Background

### 2.1. The extended Kalman filter

Kalman Filter is often used when a state of a system cannot be measured directly [27]. To estimate the state of the system optimally, Kalman Filter is used. In the case of multiple sensors with noise, Kalman Filter can be used to combine data from sensors and make an estimation of the signal that cannot be directly measured. This is called sensor fusion [31]. Position estimation methods, inertia measurement unit, navigation and guidance, aviation monitoring, vehicle control, statistics and economics are a few of the major areas where the Kalman filter is used [22]. In multi-sensor data fusion, the Kalman Filter is of great importance.

Since the Kalman Filter is based on the Gaussian distribution, the Gaussian distribution is preserved after a linear transformation [31]. Nonetheless, the problem in this article and most of the practical problems are not linear. The output of a nonlinear function does not have a Gaussian distribution if Gaussian distribution is applied to the function [26]. Accordingly, when making an estimation of a nonlinear function, the Extended Kalman Filter (EKF) is used. The Extended Kalman Filter becomes applicable to nonlinear functions by linearizing the system model around the estimation of the mean of the current state. The linearization is made by using Taylor series expansion [28]. To get an approximation of the estimation, locally linearized model is used [26]. Navigation systems is one of the numerous

real-time applications where Extended Kalman Filters are used. Compared to other non-linear filtration methods such as particle filters and point-mass filters, the Extended Kalman Filter is less expensive in terms of computational cost [17].

Kalman filters predicts next states of the system, considering of the current noise characteristics and the previous states of the system. The transition between the states of  $k-1$  and  $k$  is given as:

$$x_k = Fx_{k-1} + Bu_{k-1} + w_{k-1}. \quad (1)$$

In equation (1),  $x$  represents the state vector,  $F$  represents the state transition matrix,  $B$  represents control matrix of the input  $u$  and  $w$  represents the zero mean Gaussian process noise [17].

State transition and measurement models for EKF are given as:

$$x_k = f(x | k-1, u_{k-1}) + w_{k-1}, \quad (2)$$

$$z_k = h(x_k) + v_k, \quad (3)$$

where  $f$  is the function of  $x_{k-1}$  and  $u_{k-1}$ , and it provides the current state  $x_{k-1}$ . The measurement is  $z_k$ , the measurement noise is  $v_k$  and  $h$  is the measurement function. There are two stages in the Extended Kalman Filter. These two stages are called update stage and prediction stage. Input of the prediction stage is the output of the update stage from previous time step. In the update stage, updated state estimates and Kalman gain are calculated with the outputs of the prediction stage. The model of prediction stage is presented as:

$$\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1}), \quad (4)$$

$$P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q, \quad (5)$$

where the estimate is denoted by hat operator  $\hat{\cdot}$ , prior is signified by '+', posterior is signified by '-', predicted state estimate is denoted by  $\hat{x}_k^-$  and predicted error covariance is denoted by  $P_k^-$  [17]. The update stage model is presented as:

$$y_k = z_k - h(\hat{x}_k^-), \quad (6)$$

$$K_k = P_k^- H_k^T (R + H_k P_k^- H_k^T)^{-1}, \quad (7)$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k y_k, \quad (8)$$

$$P_k^+ = (I - K_k H_k) P_k^-, \quad (9)$$

where measurement residual is  $y_k$ , Kalman gain is  $K_k$ , updated state estimate is  $\hat{x}_k^+$  and updated error covariance is  $P_k^+$  [17]. Jacobian matrices of  $f$  and  $h$  are  $F$  and  $H$ .  $R$  represents the covariance matrix of measurement noise and  $Q$  represents the covariance matrix of process noise. More information in-depth about the subject can be found in [17].

## 2.2. The nonlinear programming

Nonlinear programming tackles optimization problems where either the objective function or the constraints (or both) exhibit nonlinear behavior. Unlike linear programming, which deals with optimizing linear objective functions within linear constraints, nonlinear programming extends this approach to more complex mathematical models.

In nonlinear optimization, the objective is to minimize or maximize a nonlinear objective function while satisfying a set of constraints. These constraints can be either inequality or equality constraints, adding complexity to the optimization process [24]]. To solve nonlinear problems with constraints, there are numerous methods, the choice of which depends on the type of nonlinearity of the objective function and the constraints. Examples of the methods are interior-point, sequential quadratic programming (SQP) and trust-region reflective [24]. Large-scale nonlinear optimization problems with sparseness or structure are where the interior-point method is particularly useful. Sequential Quadratic Programming (SQP), which is used as a general solution to nonlinear problems, takes constraints into account at each iteration [24]. The trust-region reflective method is used for solving linear and nonlinear problems where constraints are the bounds only [24].

Nonlinear programming applied in this study is modeled as following:

$$f(X), \tag{10}$$

$$LB \leq X \leq UB, \tag{11}$$

$$A_{eq}X = b_{eq}, \tag{12}$$

$$AX \leq b, \tag{13}$$

$$C_{eq}(X) = 0, \tag{14}$$

$$C(X) \leq 0. \tag{15}$$

In equation (10), objective function is denoted by  $f(X)$ . In the nonlinear programming method, the objective function  $f(X)$  is tried to be minimized by considering the given constraints. In equation (11), upper and lower boundaries of the input  $X$  are denoted by  $UB$  and  $LB$ . This means that at each iteration, the minimum and maximum values that  $X$ , the input value of the objective function  $f(X)$ , can take are determined by  $LB$  and  $UB$ . In equation (12)  $A_{eq}$  and  $b_{eq}$  are linear equality constraints.  $A_{eq}$  and  $b_{eq}$  determine the restrictions in form of linear equality equations for the  $X$ . In equation (13), linear equality constraints are represented by  $A$  and  $b$ .  $A$  and  $b$  determine the restrictions in form of linear inequality equations for the  $X$ . In equation (14),  $C_{eq}(X)$  is nonlinear equality function.  $C_{eq}(X)$  determines the restrictions in form of nonlinear equality function

for the  $X$ . In equation (15),  $C(X)$  is nonlinear inequality function [11].  $C(X)$  determines the restrictions in form of nonlinear inequality function for the  $X$ .

### 2.3. Homogeneous transformation matrices

The transformation of a point to another point in three-dimensional space can be expressed with the homogeneous transformation matrix [18, 29]. The three-dimensional transformation matrix consists of the rotation matrix and the transformation vector. A homogeneous transformation matrix in 3D space is  $4 \times 4$  dimensional. For example, let's take the rotation matrix  $R(\alpha, \beta, \gamma) = Rz(\alpha) Ry(\beta) Rz(\gamma)$  and the  $x_t, y_t, z_t$  translation and create a homogeneous transformation matrix. The obtained  $T$  homogeneous transformation matrix is seen in equation (16):

$$T = \begin{pmatrix} \cos \alpha \cos \beta & MA & MC & x_t \\ \sin \alpha \cos \beta & MB & MD & y_t \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma & z_t \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (16)$$

where  $MA = \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma$ ,  $MB = \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma$ ,  $MC = \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma$ ,  $MD = \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma$ .

In the  $T$  homogeneous transformation matrix in the formula, first the rotation process  $R(\alpha, \beta, \gamma)$ , and then the translation process  $x_t, y_t, z_t$  is applied. The order of operations is as follows: roll as  $\gamma$ , pitch as  $\beta$ , yaw as  $\alpha$  and translate as  $x_t, y_t, z_t$ , respectively. A point  $A$  that is transformed has six degrees of freedom, including rotation in three axes and translation in three axes [18].

### 3. The proposed method

The proposed method of this study concerns an escape path prediction algorithm using Extended Kalman Filter and Nonlinear Programming methods. To develop the proposed algorithm with necessary implementations and simulations, MATLAB is used as a tool. Figure 1 shows simplified architecture of the proposed method. The position of the friendly aircraft in the  $x, y$ , and  $z$  coordinate axes to which the escape path prediction algorithm will be applied, is denoted by  $p\_ally$ . Position of the enemy aircraft in the  $x, y, z$  coordinates is denoted by  $p\_enemy$ . Positions  $p\_est$  (enemy aircraft's position obtained from the Extended Kalman Filter) and  $p\_NLP$  (optimal escape vector) are evaluated for each time step in a loop, in each time interval.

MATLAB's built-in function for nonlinear constraint optimization is called "fmincon". The estimation value  $p\_est$  is sent to the fmincon function as input [11]. The optimal escape vector  $p\_NLP$  comes as an output of the fmincon

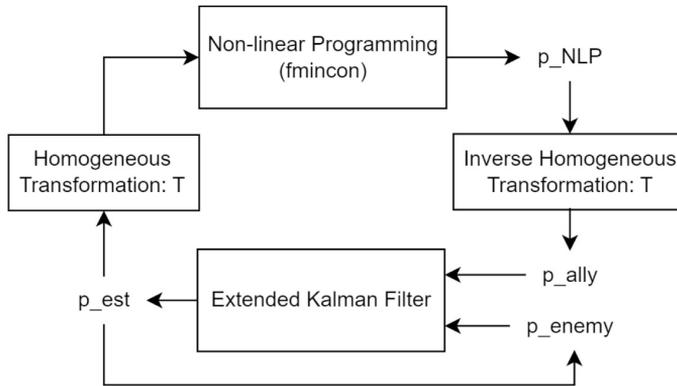


Figure 1: The proposed method system architecture

function. The next position of  $p\_ally$  is obtained by adding the vector  $p\_NLP$  to the previous position of  $p\_ally$ . In this way, an optimal escape route for the friendly aircraft is determined based on the attacking aircraft’s position estimation. As the different maneuvers are performed by the ally UAV at each time step, the ally UAV’s coordinate axes changes. Therefore, solution and the nonlinear constraints must be in the ally UAV’s coordinate system at each time step. Before passing the relative distance between the attacking UAV and the ally UAV to the nonlinear programming method `fmincon`, it must undergo a homogeneous transformation to the ally UAV’s coordinate system from the original coordinate system. The escape vector obtained as the optimal solution undergoes an inverse homogeneous transformation. This transformation is made to transfer the escape vector to the original coordinate system before being applied to the allied UAV.

The Extended Kalman Filter model implemented in this study is convenient for the scenarios selected. As mentioned in the previous section, compared to other non-linear filtration methods such as particle filters and point-mass filters, the Extended Kalman Filter is less expensive in terms of computational cost [17]. In this model, two sensors on the allied aircraft are used: the angle sensor and the range sensor. Attacking aircraft’s position is estimated using the Extended Kalman Filter, using the angle and distance data coming from these two sensors on the allied aircraft under the noise [17]. Standard deviation of measurement noise  $v_k$  and standard deviation of process noise  $w_{k-1}$  are passed as inputs to the Extended Kalman Filter. Hence, the EKF can perform and be tested under different noise characteristics as desired. A detailed Extended Kalman Filter diagram can be seen in Fig. 2.

Starting positions of the enemy and allied aircrafts has been set in the initialization stage. Predictions of state estimate and error covariance has been made

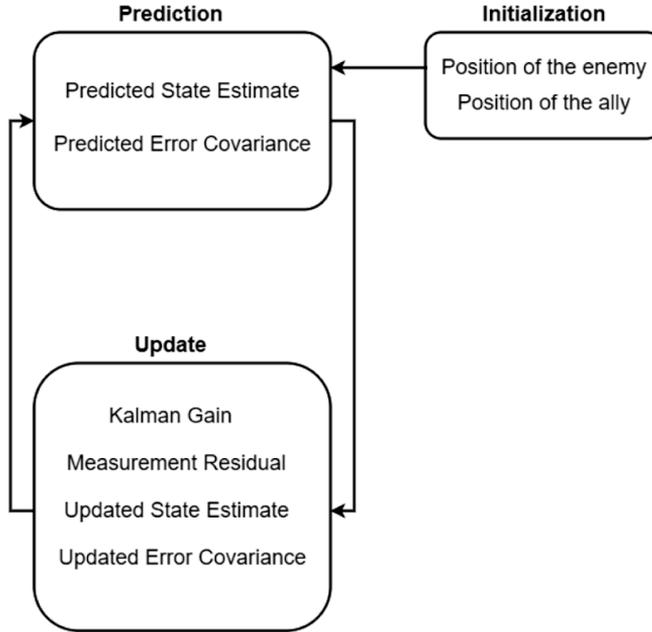


Figure 2: The extended Kalman filter model

as in equations (4) and (5), in the prediction stage. Measurement residual and Kalman gain are evaluated as in equations (6) and (7), in the update stage. State estimate and error covariance are also updated as in equations (8) and (9). In the Extended Kalman Filter model, the radar distance sensor and angle sensor of the ally UAV measure the distance and angle values of the enemy UAV. The distance sensor measures the distance  $r$  between two planes of the spherical coordinate system shown in Fig. 3. The angle sensor measures the azimuth  $\varphi$  and polar  $\theta$  angles shown in Fig. 3. EKF estimates the position of the enemy.

Fmincon function solves minima problems of constrained multivariable functions. As described in equations (10), (11), (12), (13), (14) and (15), the constraints for the objective function and the mentioned function are defined for fmincon. Fmincon begins from an initial point  $x_0$  and goes to the  $x$  value which minimizes the function with defined constraints [11]. The main use of fmincon is to find a minimizing value for an objective function. In this proposed method, a basic score function, which is frequently used in aerospace and aviation fields, is used as the objective function. It is used to evaluate the relative distance and collision between two aircrafts [4, 21]. The objective score function is defined as [25]:

$$S_c = \left(1 - \frac{|\mathcal{E} + \lambda|}{\pi}\right) e^{-\frac{d-d_{\text{opt}}}{K\pi}}. \quad (17)$$

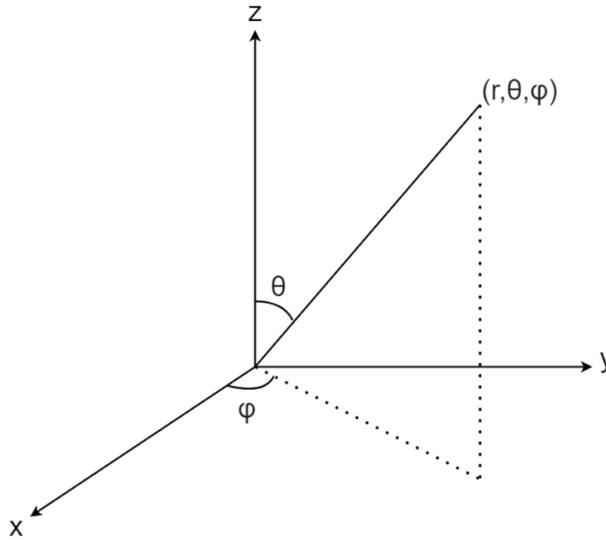


Figure 3: The spherical coordinate system

In equation (17),  $S_c$  represents the score value obtained from angle and position of the two aircraft relative to each other.  $\varepsilon$  and  $\lambda$  represent the angles between the LOS (Line of Sight) line and the movement vectors of two aircraft, respectively. The  $\varepsilon$  and  $\lambda$ , relative angles of two aircrafts are described in Fig. 4. Distance between the two aircrafts is represented by  $d$ . Desired optimal distance value is represented by  $d_{opt}$ . Lastly, proportional adjustment between the distance and angle is made using the  $K$  constant. When determining the values of  $K$  and  $d_{opt}$  parameters, values of 600 for  $K$  and 700 for  $d_{opt}$  were chosen, as used in a similar study [25]. This choice is based on both established practices in the literature and experimental findings aimed at improving the accuracy of the method. By adopting these values, the method is expected to provide comparable results and facilitate appropriate generalization. For these reasons,  $K$  and  $d_{opt}$  variables are fixed to these values. An example design in which these values are also variable can be designed in future studies.

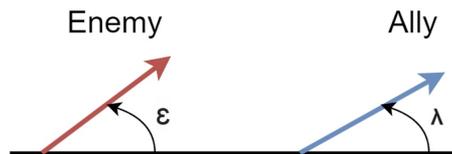


Figure 4: Relative angles of the two aircrafts (adapted from [25])

In the `fmincon` function used in the constrained optimization, linear inequality constraints are used to give the escaping UAV a conical angle. In addition, non-linear inequality constraints are applied to constrain the resultant velocity vector of the escaping UAV. These constraints will be discussed later in this section. Angular constraints have been applied to the constrained optimization function `fmincon` to ensure that the escaping plane's motion vector stays inside the circular sector with a desired angle. These constraints are applied for the angles of the  $x$ -axis of the motion vector with the  $y$  and  $z$  axes. Figure 5 shows the motion vector remaining inside the representative circular sector.

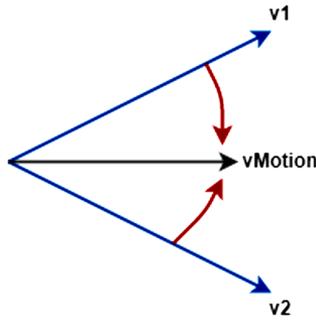


Figure 5: Motion vector inside the circular sector

It is aimed to ensure that  $vMotion$  (Fig. 5), stays between  $v1$  and  $v2$  vectors representing a circular sector with the desired angle. To do so, it is necessary to determine whether the motion vector  $vMotion$  stays clockwise relative to the  $v1$  vector and counterclockwise relative to the  $v2$  vector. To determine whether the  $vMotion$  vector remains clockwise relative to the  $v1$  vector, the normal vector of the  $v1$  vector is calculated and the dot product of the  $vMotion$  vector and the normal vector is taken. If the result from the dot product is positive, the  $vMotion$  vector is located counterclockwise relative to the  $v1$  vector. If the result is negative, the  $vMotion$  vector is located clockwise relative to the  $v1$  vector. The same procedure is applied to the  $v2$  vector. With this method, the matrices required to apply the angular constraints to `fmincon` are determined. These matrices are the  $A$  and  $b$  matrices in equation (13).  $A$  and  $b$  matrices seen in equation (13) are given to `fmincon` as in equations (18) and (19), according to the angular constraint finding method described in this section.

$$A = \begin{bmatrix} -MRV * \sin \alpha & MRV * \cos \alpha & 0 \\ -MRV * \sin \alpha & -MRV * \cos \alpha & 0 \\ -MRV * \sin \beta & 0 & MRV * \cos \beta \\ -MRV * \sin \beta & 0 & -MRV * \cos \beta \end{bmatrix}, \quad (18)$$

$$b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (19)$$

In equation (18), MRV is the maximum resultant velocity constraint of input  $X = x, y, z$  in terms of m/s. The angle  $\alpha$  is the yaw angle between the  $x$  and  $y$  axis. The angle  $\beta$  is the pitch angle between the  $x$  and  $z$  axis.

To represent the maximum speed the escaping UAV can reach, a resultant velocity constraint to be applied in `fmincon` is given. This is a constraint in form of non-linear inequality shown in the following equation.

$$C(X) = X(1)^2 + X(2)^2 + X(3)^2 - (\text{MRV})^2 \leq 0, \quad (20)$$

where  $X$  is the input of the coordinates  $x, y, z$  of the escaping UAV. In the simulations MRV is set to 277. The value 1000 is the maximum velocity of the escaping UAV in terms of km/h. It is desired to convert km/h to m/s since a time step in the simulation is 1 second.

After the estimation of the direction and position of the enemy aircraft, the constraint optimization is performed. As UAV's maneuverability changes according to current angle made in the pitch, the roll, and the yaw axes, constraints get determined on the coordinate system of the escaping UAV. In this way, nonlinear programming is run according to coordinate axes of the escaping UAV. Moreover, the UAV can make a different maneuver in each time step. As a result, for each time step escaping UAV's coordinate axes get altered. The original coordinate axes and the coordinate axes of the escaping UAV should be carefully considered in the algorithm. Therefore, the solution requires rotating and translating the optimal UAV position and direction obtained from nonlinear programming to the original coordinate axes from the coordinate axes of the escaping UAV. This algorithm uses a method similar to the one described in [10].

A flight mechanism must be taken into account to determine the UAV's direction. In this article, the approach used for UAV maneuverability is selected as in [13]. There are three axes of the UAV: the pitch, the roll, and the yaw. Elevators are used to change the altitude (alter the pitch). After achieving the desired altitude, the UAV updates its position as flat to the ground. As a result, during the optimization process, the  $z$  axis in the UAV and the original coordinate  $z$  axis are identical. However, in addition to  $z$  axis movement,  $x$  and  $y$  axes motions are also considered while moving to various positions in three-dimensional space. It is presumable that the UAV has a great maneuvering capacity. Here, the following flight mechanism is considered when turning right or left: Instead of utilizing the rudder to change the yaw angle, roll control is accomplished by first using

the ailerons or flaperons, followed by the elevator, and the targeted direction is determined with the aircraft's nose. The UAV adjusts its position such that it is flat to the ground after moving in the desired direction in the  $x$ - $y$  plane.

In summary, based on the two methods; EKF and Nonlinear Programming and flight mechanism, an escape path prediction algorithm is realized.

#### 4. Simulations and results

In this section, the simulation scenarios are explained, and the test results are examined. The simulation results are realized using MATLAB. It has been assisted from [14] while implementing the Extended Kalman Filter for simulation of these scenarios. The following is a description of the scenarios: There are two UAVs flying in the open space, one of them is the escaping allied UAV and the hostile one is enemy UAV. The enemy UAV's initial point is  $(0, 0, 0)$  meters in  $x$ ,  $y$ , and  $z$  coordinates and just along the  $x$ -axis, it moves at 1000 km/h. The ally UAV's initial point is  $(500, 0, 0)$  meters in  $x$ ,  $y$ , and  $z$  coordinates. The radar system with angle and range sensors is the source of input of the ally UAV for target tracking. For velocity in all three dimensions, the process noise's standard deviation is set at 0.5 m/s. The selected simulation scenarios were run for a 30 second period.

The upper and lower boundaries for input  $X$  in nonlinear programming are applied as follows for  $x$ ,  $y$ , and  $z$ :  $UB = [277.8, 69.4, 69.4]$ ,  $LB = [-69.4, -69.4, -69.4]$  in meters. The solution of the `fmincon` can have a range of values between  $UB$  and  $LB$ , which are expressed in meters per second.  $UB$  and  $LB$  were chosen according to the mobility of the ally UAV and were calculated by unit converting 1000 km/h to m/s in  $x$  axis and 250 km/h to m/s in  $y$  and  $z$  axes. The highest consequent value for the solution of `fmincon` in the combined  $x$ ,  $y$ , and  $z$  axes is the nonlinear inequality function  $C(X)$ . Equation (20) is used to compute  $C(X)$ , where 1000 is the maximum velocity in km/h and 0.28 is the constant used to convert km/h to m/s. In the last three scenarios, linear inequality constraints are used as in 7 (13) to constrain the movement angle of the motion vector in a desired angle. The used linear inequality constraints are equal to the expression in equations (18) and (19). Values of the  $\alpha$  and the  $\beta$  in equation (18) are given  $\pi/9$  radians.

The first scenario has run on MATLAB, and the standard deviation of measurement noise input was set to  $[0.1, 0.1, 5]$  for the angles and distance, respectively. Construction of the measurement covariance matrix has made applying noise characteristics as  $[0.5, 0.5, 10]$ . Figure 6 presents the  $xy$ -axis trajectories of the ally UAV and the enemy UAV. Figure 7 presents the three-dimensional trajectories of UAVs. In the figures where the trajectories are shown, unit of the

values is meter. As seen in Figs. 6 and 7, the ally UAV escaped by deviating in the  $y$  and  $z$  axes along the escape path. As the enemy UAV's estimated position sways over time in the  $z$  axis, the ally UAV slopes in the  $z$  axis in positive direction

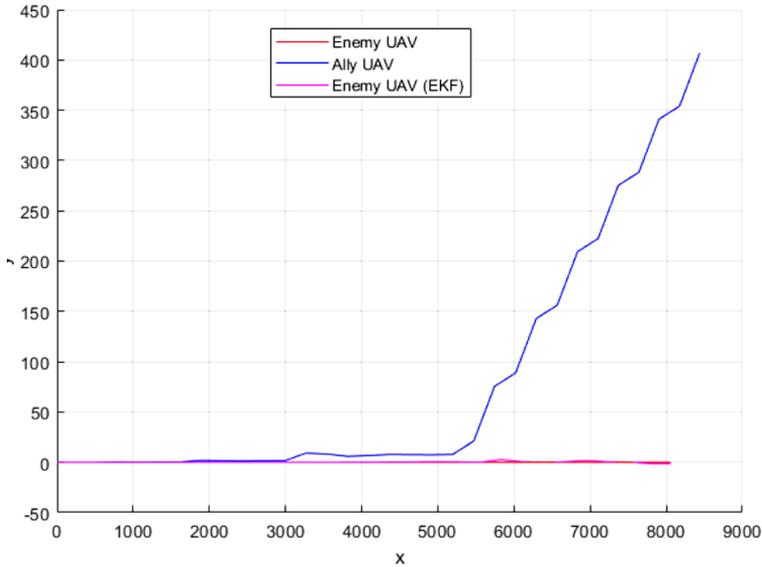


Figure 6: Trajectories of the UAVs in the Scenario 1 in  $xy$ -axis

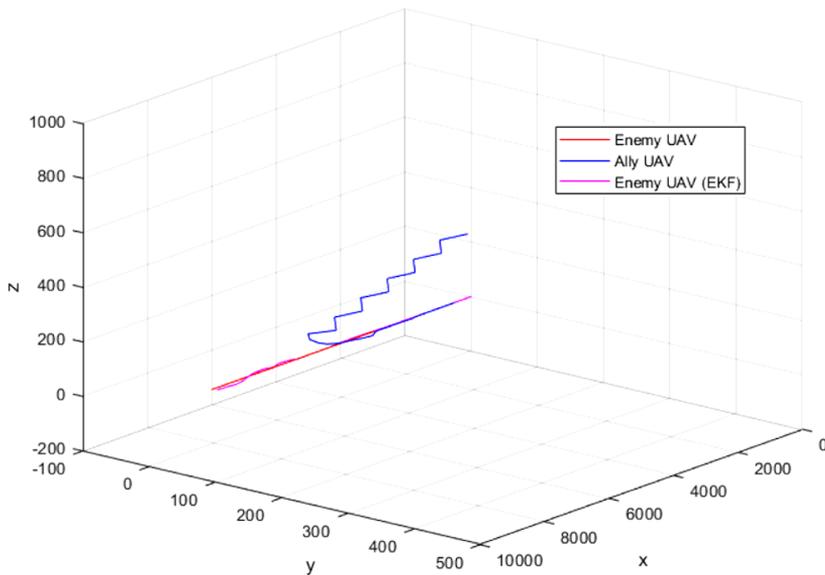


Figure 7: Trajectories of the UAVs in the Scenario 1 in 3D

and escapes. Since the measurement noise characteristic values are low, ally UAV escaping by drawing the trajectory close to the ground truth. In this scenario, the angular constraints are not used like the method given in [13].

The second scenario has run on MATLAB, and the standard deviation of measurement noise input was set to  $[0.1, 0.1, 5]$  for the angles and the distance respectively. Construction of the measurement covariance matrix has made applying noise characteristics as  $[0.5, 0.5, 10]$ . The trajectories in the  $xy$ -axis and the 3D-axis of enemy UAV and ally UAV are shown in Fig. 8 and Fig. 9. The input of standard deviation of measurement noise of this scenario is the same as in the first scenario. The difference between this scenario and the first one is that angular constraints are applied to `fmincon` as linear inequalities. With the application of angular constraints, the movements of the escaping UAV for each time step are more constrained than in the first scenario. The turns of the escaping UAV for each time step are also limited by the angles determined in these constraints.

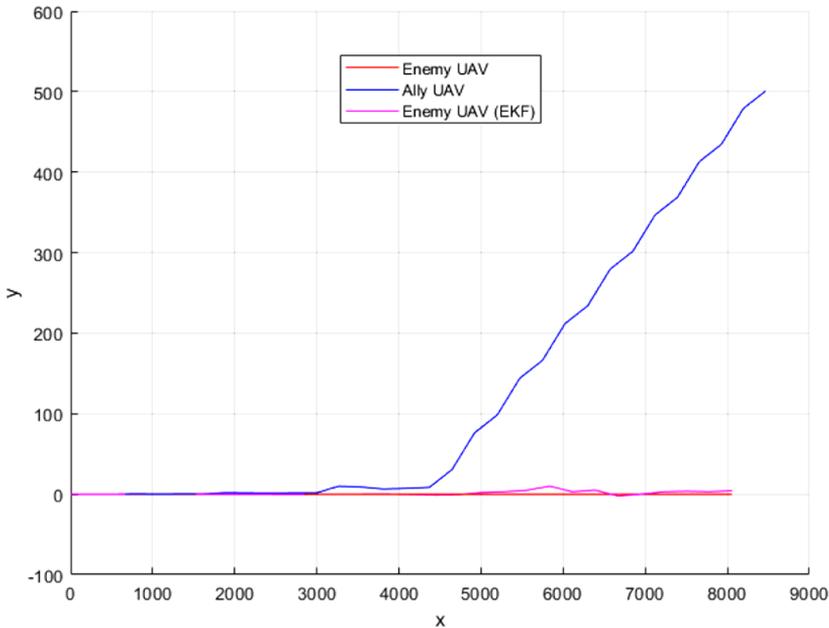


Figure 8: Trajectories of the UAVs in the Scenario 2 in  $xy$ -axis

In the third scenario, the standard deviation of measurement noise input was set to  $[0.5, 0.5, 10]$  for the angles and the distance respectively. The trajectories in the  $xy$ -axis and the 3D-axis of enemy UAV and ally UAV are shown in Fig. 10 and Fig. 11. The turns of the escaping UAV for each time step are also limited by the angles determined in these constraints.

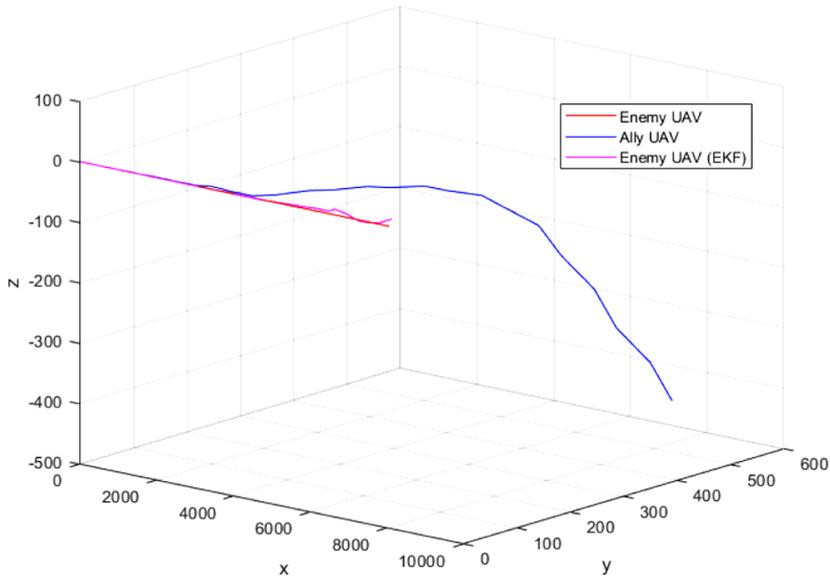


Figure 9: Trajectories of the UAVs in the Scenario 2 in 3D

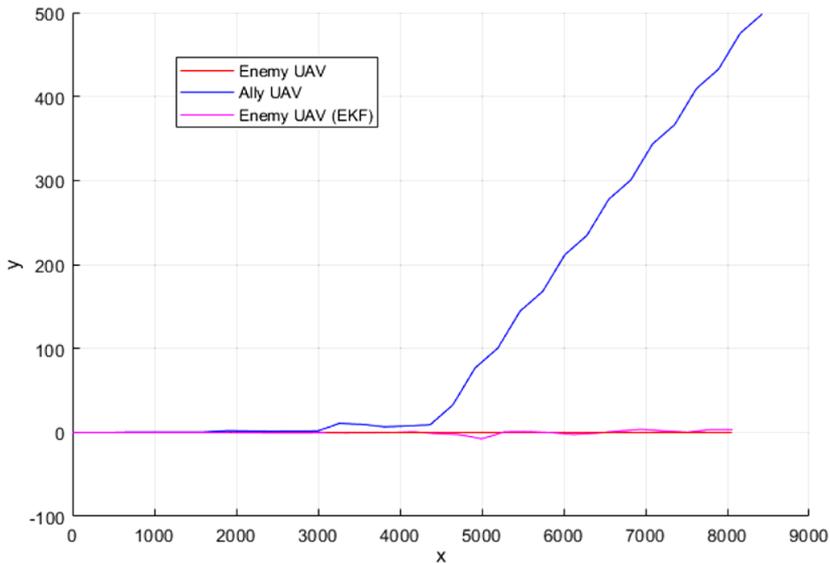


Figure 10: Trajectories of the UAVs in the Scenario 3 in  $xy$ -axis

In the fourth scenario, the standard deviation of measurement noise input was set to  $[1.5, 1.5, 50]$  for the angles and the distance respectively. The trajectories in the  $xy$ -axis and the 3D-axis of enemy UAV and ally UAV are shown in Fig. 12 and Fig. 13.

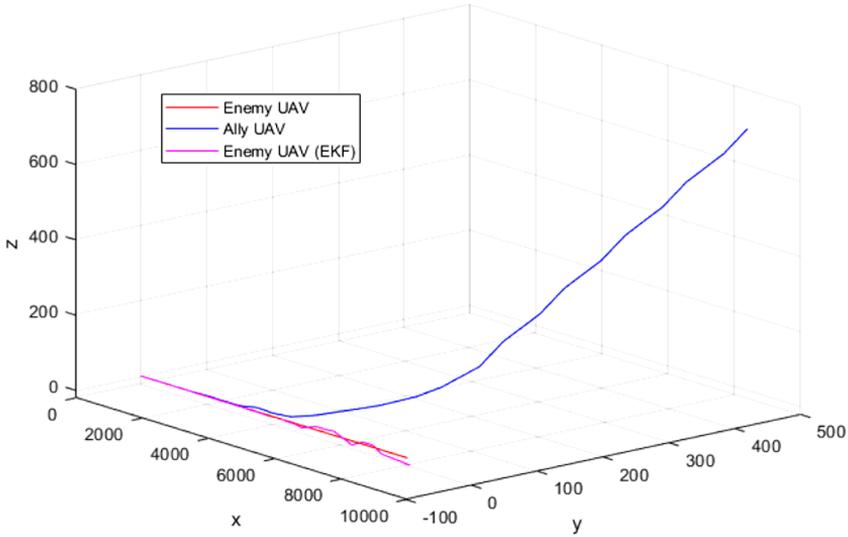


Figure 11: Trajectories of the UAVs in the Scenario 3 in 3D

The ground truth calculation is made by calculating the escape route for the ally UAV according to the actual position of the enemy UAV, not according to the estimated position on which EKF was applied. In other words, constrained optimization method NLP is applied based on real trajectory of the enemy UAV.

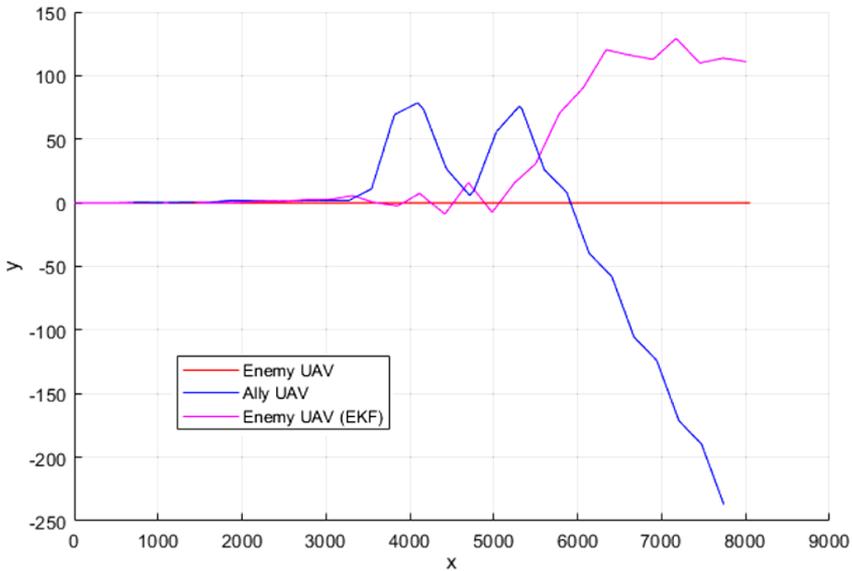


Figure 12: Trajectories of the UAVs in the Scenario 4 in xy-axis

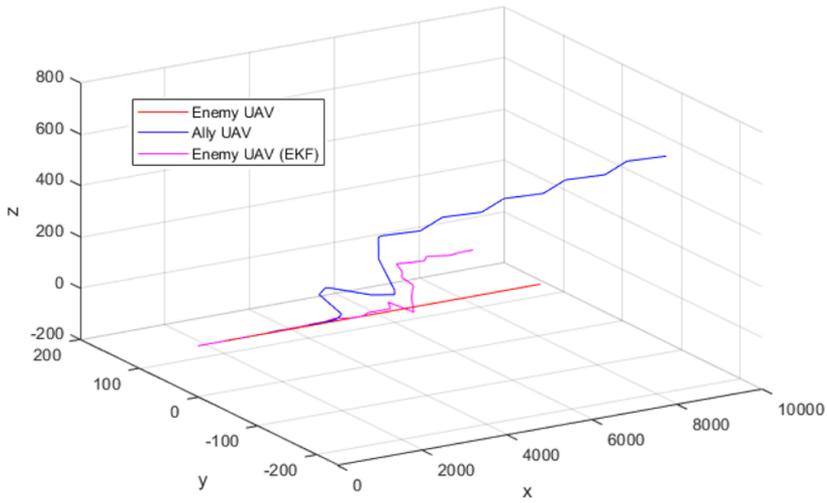


Figure 13: Trajectories of the UAVs in the Scenario 4 in 3D

In order to make a comparison with the allied UAV's escape trajectories that planned according to estimated enemy UAV positions by applying EKF, ground truth version of the allied UAV's escape trajectory can be seen in Fig. 14 and Fig. 15.

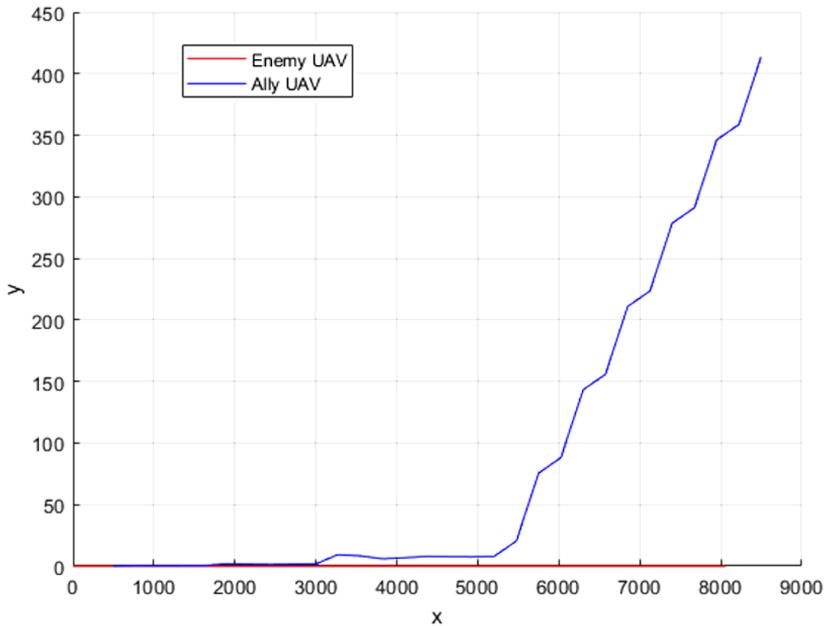


Figure 14: Trajectories of the UAVs in the ground truth in  $xy$ -axis

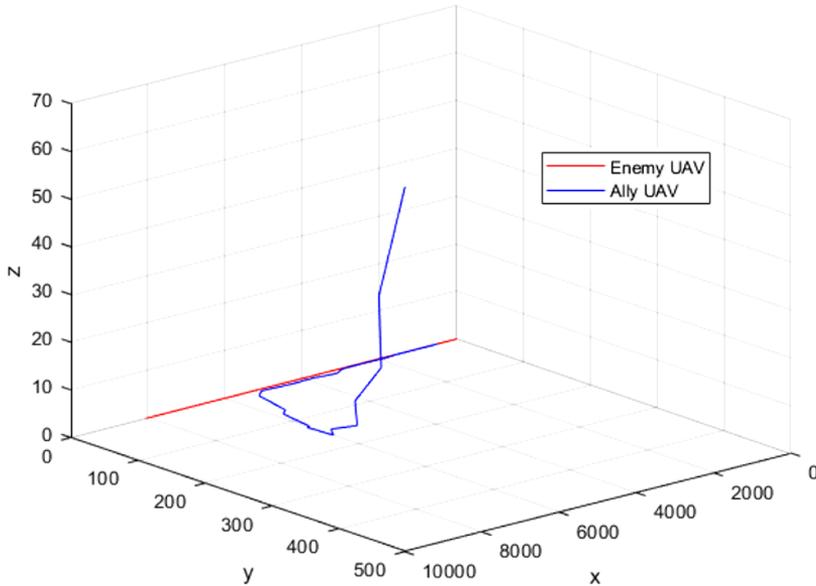


Figure 15: Trajectories of the UAVs in the ground truth in 3D

## 5. Conclusion

In this work, an algorithm for predicting escape paths that combines the Extended Kalman Filter and Nonlinear Programming using maneuverability angle constraints is presented. Extended Kalman Filter is used as the estimation method since the Extended Kalman Filter is less expensive in terms of computational cost compared to other non-linear filtration methods such as point-mass filters and particle filters [17]. Nonlinear Programming is used for the constraint optimization. Coordinate system transformations are made for the UAV positions by using homogeneous transformation matrices. Distance and angle measurements coming from allied UAV's sensors and standard deviation values for measurement noise are used as inputs for Extended Kalman Filter. Thus, a sensor fusion and escape path planning method has been developed. The algorithm has been implemented on MATLAB, and the simulations for different scenarios has been tested. A ground truth of the escape path of allied UAV is calculated. In the ground truth, escape path planned by performing constrained optimization based on real position of the enemy UAV, not the estimations obtained from Extended Kalman Filter.

It can be seen from the simulation results that varying measurement noise standard deviation values produce various EKF outcomes. Therefore, nonlinear programming comes with different solutions. For lower values of standard devi-

ation of measurement noise, escape path of the allied UAV becomes more alike with the ground truth. For higher values, escape path of the allied UAV becomes more inefficient in terms of results and resource use. As is clear, the effectiveness of the radars in UAVs can be a crucial component of artificial intelligence-based algorithms like the one created in this work. Additionally, the results of scenarios with angular constraints applied and the scenario (the first scenario) without angular constraints applied are compared. It has been observed that, with angular constraints applied considering a desired angle, the UAV model has more realistic constraints and moves closer to reality. However, the model used in this study does not have enough inputs to make the inference to determine the end of the engagement. Determining the end of the engagement depends on factors such as the effect of the distance between the two aircraft on the measurements from the radars, the type of enemy, the type of engagement. Besides of these, the most important factor in determining the termination of engagement is knowing the type of missile on the enemy aircraft and how the enemy aircraft will use the missiles on it. Since the enemy missiles are not modeled in this study, the determination of the end of the engagement is not made. However, this process may be added in future studies. For future studies, more efficient and more flexible Extended Kalman Filter models for various situations can be developed. Also, a system model can be developed in which enemy missiles are analyzed and modeled.

## References

- [1] Anka – tusas, www.tusas.com. [Online]. Available: <https://www.tusas.com/en/products/uav/operative-strategic-uav-systems/anka> (visited on 01/05/2022).
- [2] S. BORTOFF: Path planning for UAVs. *IEEE Xplore*, (2000). DOI: [10.1109/acc.2000.878915](https://doi.org/10.1109/acc.2000.878915)
- [3] S.P. BRADLEY, A.C. HAX and T.L. MAGNANTI: *Applied Mathematical Programming*. Addison-Wesley. 1977.
- [4] G.H. BURGIN, A.J. OWENS and A. AND: *I. Decision Science, An adaptive maneuvering logic computer program for the simulation of one-on-one air-to-air combat Vol. 2: Program description*. NTIS. 1975.
- [5] F. CAPPELLO, R. SABATINI, S. RAMASAMY and M. MARINO: Particle filter-based multi-sensor data fusion techniques for RPAS navigation and guidance. *IEEE Metrology for Aerospace (MetroAeroSpace)*, (2015). DOI: [10.1109/metroaerospace.2015.7180689](https://doi.org/10.1109/metroaerospace.2015.7180689)
- [6] H. CHEN, X. WANG and Y. LI: A survey of autonomous control for UAV. *International Conference on Artificial Intelligence and Computational Intelligence*, Shanghai, China, (2009). DOI: [10.1109/aici.2009.147](https://doi.org/10.1109/aici.2009.147)
- [7] S.Y. CHOI AND D. CHA: Unmanned aerial vehicles using machine learning for autonomous flight; state-of-the-art. *Advanced Robotics*, **33**(6), (2019), 265–277. DOI: [10.1080/01691864.2019.1586760](https://doi.org/10.1080/01691864.2019.1586760)

- [8] A. DOBRESCU, M. OLTEAN and G. UDEANU: Unmanned aerial vehicle in military operations. *Scientific Research and Education in The Air Force*, **18** (2016), 199–206. DOI: [10.19062/2247-3173.2016.18.1.26](https://doi.org/10.19062/2247-3173.2016.18.1.26)
- [9] T.N. DUPUY: *Understanding War: History and Theory of Combat*. London Leo Cooper. 1992.
- [10] Expressing Points in Different Coordinate Systems, [www.dgp.toronto.edu](http://www.dgp.toronto.edu). [Online]. Available: <http://www.dgp.toronto.edu/~neff/teaching/418/transformations/transformation.html>
- [11] Find Minimum of Constrained Nonlinear Multivariable Function – Matlab fmincon, [www.mathworks.com](http://www.mathworks.com). [Online]. Available: <https://www.mathworks.com/help/optim/ug/fmincon.html>
- [12] D. GLADE and L. COL: Unmanned aerial vehicles: Implications for military operations. [Online]. Available: <https://apps.dtic.mil/sti/pdfs/ADA425476.pdf>
- [13] E.N. GÖKAL and U. SAKARYA: Multi-sensor data fusion for path prediction of escaping from engagement in unmanned aerial vehicle scenario. *European Journal of Science and Technology*, **32** (2021), 705–710. DOI: [10.31590/EJOSAT.1039358](https://doi.org/10.31590/EJOSAT.1039358)
- [14] Introduction to Kalman Filter and Its Applications. [www.mathworks.com](http://www.mathworks.com). [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/68262-introduction-to-kalman-filter-and-itsapplications>
- [15] H. JIANG and Y. LIANG: Online path planning of autonomous UAVs for bearing-only standoff multi-target following in threat environment. *IEEE Access*, **6** (2018), 22531–22544. DOI: [10.1109/access.2018.2824849](https://doi.org/10.1109/access.2018.2824849)
- [16] M. KANG, Y. LIU and Y. ZHAO: A threat modeling method based on Kalman filter for UAV path planning. *29th Chinese Control and Decision Conference*, (2017). DOI: [10.1109/c-cdc.2017.7979170](https://doi.org/10.1109/c-cdc.2017.7979170)
- [17] Y. KIM and H. BANG: Introduction and Implementations of the Kalman Filter. In *Introduction and implementations of the Kalman filter*. IntechOpen, F. Govaers (Ed.), 2018. DOI: [10.5772/intechopen.80600](https://doi.org/10.5772/intechopen.80600)
- [18] S.M. LAVALLE: *Planning Algorithms*. Cambridge University Press. 2006.
- [19] C. LUO, S.I. MCCLEAN, G. PARR, L. TEACY and R. DE NARDI: UAV position estimation and collision avoidance using the extended Kalman filter. *IEEE Transactions on Vehicular Technology*, **62** (2013), 2749–2762. DOI: [10.1109/tvt.2013.2243480](https://doi.org/10.1109/tvt.2013.2243480)
- [20] G. MAO, S. DRAKE and B.D.O. ANDERSON: Design of an extended Kalman filter for UAV localization. *Information, Decision and Control*, (2007). DOI: [10.1109/idc.2007.374554](https://doi.org/10.1109/idc.2007.374554)
- [21] J.S. MCGREW, J.P. HOW, B. WILLIAMS and N. ROY: Air-combat strategy using approximate dynamic programming. *Journal of Guidance, Control, and Dynamics*, **33** (2010), 1641–1654. DOI: [10.2514/1.46815](https://doi.org/10.2514/1.46815)
- [22] R.J. MEINHOLD and N.D. SINGPURWALLA: Understanding the Kalman filter. *The American Statistician*, **37** (1983), 123–127. DOI: [10.1080/00031305.1983.10482723](https://doi.org/10.1080/00031305.1983.10482723)
- [23] Z. ZHANG, J. LI and J. WANG: Sequential convex programming for nonlinear optimal control problems in UAV path planning. *Aerospace Science and Technology*, **76** (2018), 280–290. DOI: [10.1016/j.ast.2018.01.040](https://doi.org/10.1016/j.ast.2018.01.040)

- [24] Nonlinear Programming. [www.mathworks.com](http://www.mathworks.com). [Online]. Available: <https://www.mathworks.com/discovery/nonlinear-programming.html>
- [25] N. RAMÍREZ LÓPEZ and R. ZBIKOWSKI: Effectiveness of autonomous decision making for unmanned combat aerial vehicles in dogfight engagements. *Journal of Guidance, Control, and Dynamics*, **41** (2018), 1021–1024. DOI: [10.2514/1.g002937](https://doi.org/10.2514/1.g002937)
- [26] M.I. RIBEIRO: Kalman and extended Kalman filters: Concept, derivation and properties. (2004). [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.214.809&rep=rep1&type=pdf>
- [27] D. SIMON: Kalman filtering. *Embedded Systems Programming*, **14**(6), (2001), 72–79.
- [28] D.J. SIMON: Using nonlinear Kalman filtering to estimate signals. *Embedded Systems Design*, **19**(7), (2006), 38–53. DOI: [10.1115/1.1789153](https://doi.org/10.1115/1.1789153)
- [29] T.M. STRAT: Recovering the camera parameters from a transformation matrix. *Readings In Computer Vision*, (1987), 93–100. DOI: [10.1016/b978-0-08-051581-6.50017-9](https://doi.org/10.1016/b978-0-08-051581-6.50017-9)
- [30] J. TISDALE, Z. KIM and J. HEDRICK: Autonomous UAV path planning and estimation. *IEEE Robotics and Automation Magazine*, **16**(2), (2009), 35–42. DOI: [10.1109/mra.2009.932529](https://doi.org/10.1109/mra.2009.932529)
- [31] Understanding Kalman Filters. [www.mathworks.com](http://www.mathworks.com). [Online]. Available: <https://www.mathworks.com/videos/series/understanding-kalman-filters.html>
- [32] Z. WU, J. LI, J. ZUO and S. LI: Path planning of UAVs based on collision probability and Kalman filter. *IEEE Access*, **6** (2018), 34237–34245. DOI: [10.1109/access.2018.2817648](https://doi.org/10.1109/access.2018.2817648)
- [33] Q. YANG, J. ZHANG and G. SHI: Path planning for unmanned aerial vehicle passive detection under the framework of partially observable Markov decision process. *Chinese Control and Decision Conference*, (2018). DOI: [10.1109/ccdc.2018.8407800](https://doi.org/10.1109/ccdc.2018.8407800)